The pwpfit Toolbox for Polynomial and Piece-wise Polynomial Data Fitting

Torbjørn Cunis

The pwpfit Toolbox for Polynomial and Piece-wise Polynomial Data Fitting

Torbjørn Cunis [*,**]

[*] *Department of Information Processing and Systems, ONERA – The French Aerospace Lab, Centre Midi-Pyrénées, 31055 Toulouse, France (e-mail: torbjoern.cunis@onera.fr)*
[**] *Drones Research Group, French Civil Aviation School, 31055 Toulouse, France (e-mail: torbjoern.cunis@recherche.enac.fr)*

Abstract: Several techniques have been proposed for piece-wise regression as extension to standard polynomial data fitting, either selecting the joints a priori or adding computational load for optimal joints. The `pwpfit` [1] toolbox provides piece-wise polynomial fitting without pre-selection of joints using linear-least square (LSQ) optimization only. Additional constraints are realised as constraint matrices for the LSQ problem. We give an application example for the multi-variable aerodynamic coefficients of the general transport model in pre-stall and post-stall.

Keywords: Grey Box Modeling; Toolboxes; Mechanical and Aerospace; Multivariable System Identification; Nonlinear System Identification; Hybrid System Identification.

1. INTRODUCTION

Polynomial data fitting names a branch of approaches dedicated to the problem of optimal coefficients for a polynomial function f, such that f approximates the measured data points, usually called "observations". A common solution consists of minimising the sum of squared residuals of f with respect to the observations using *linear least-square* (LSQ) techniques (Kariya and Kurata, 2004). Several methods exist to solve LSQ problems (Golub, 1982; Lawson and Hanson, 1995). Compared to modern tools, polynomials benefit from fast and simple evaluation.

However, a single polynomial function might not be suitable to describe the observed characteristics. Early approaches included regression of a few polynomial functions *piece-wise* over the observations; in order to find suitable switching surfaces (*joints*) for the piece-wise functions, these approaches used maximum-likelihood or Newton-Gauss methods (Robison, 1964; Gallant and Fuller, 1973), hierarchical clustering (McGee and Carleton, 1970), or regressions trees (Chaudhuri et al., 1994). Later, *multivariate splines* were introduced fitting sequences of polynomial functions over fine grids, which are rectangular (Klein and Morelli, 2006) or triangular (de Visser et al., 2009) partitions of the observations. Here, the knots of the grids, *i.e.* the joints of the piece-wise functions, are chosen prior, and are not a subject of, the fit. Both piece-wise regression and multivariate splines ensure the fitted piece-wise functions to be continuous or even smooth at their joints.

While splines today present a powerful yet complex tool for accurate and smooth interpolation, they lack of an underlying physical model justifying the partition. [2] The problem of finding appropriate joints remains open.

In this paper, we introduce the `pwpfit` [1] toolbox for MATLAB, which uses standard LSQ techniques while leaving the joint as parameter of optimization. The interface of the toolbox, on the other hand, resembles that of MATLAB's well-known `fit` function. [3] Following a study of the theoretical and implementation details, we discuss exemplary the fitting of piece-wise aerodynamic coefficients for the model of a typical airliner.

2. PRELIMINARIES

A *monomial of degree* n is a single product of powers where the exponents add up to the total degree n, without any scalar coefficient. We introduce the vector notation for a monomial $\boldsymbol{x} = (x_1, \ldots, x_m)$ in degrees $\boldsymbol{n} = (n_1, \ldots, n_m)$,

$$\boldsymbol{x}^{\boldsymbol{n}} = x_1^{n_1} \ldots x_m^{n_m}, \tag{1}$$

with the total degree $n = \|\boldsymbol{n}\|_1 = n_1 + \cdots + n_m$.

2.1 Monomials & Polynomials

Definition 1. $\boldsymbol{\mathcal{P}}_n(\boldsymbol{x})$ is the *vector of monomials* $\boldsymbol{x}^{\boldsymbol{\nu}}$ in variables $\boldsymbol{x} = (x_1, \ldots, x_m)$ with degrees $\boldsymbol{\nu} \in \mathbb{N}^m$ and total degrees $\|\boldsymbol{\nu}\|_1 \leq n$; and the number of elements in $\boldsymbol{\mathcal{P}}_n(\boldsymbol{x})$ is denoted by $\mathfrak{r}[n]$, *i.e.* $\boldsymbol{\mathcal{P}}_n \in \mathbb{R}\left[\boldsymbol{x}\right]^{\mathfrak{r}[n]}$.

While the order of monomials in $\boldsymbol{\mathcal{P}}_n(\boldsymbol{x})$ is arbitrary, we choose to have $\boldsymbol{x}^{\boldsymbol{\mu}}$ before $\boldsymbol{x}^{\boldsymbol{\nu}}$ if and only if $\|\boldsymbol{\mu}\|_1 < \|\boldsymbol{\nu}\|_1$ or $\boldsymbol{\mu}$ is reverse-lexicographically before $\boldsymbol{\nu}$ if $\|\boldsymbol{\mu}\|_1 = \|\boldsymbol{\nu}\|_1$. Defining the auxiliary vector $\boldsymbol{p}_N$ of monomials $\boldsymbol{x}^{\boldsymbol{\nu}}$ with $\|\boldsymbol{\nu}\|_1 = N$, recursively over the number of variables m as

$$\boldsymbol{p}_N(\boldsymbol{x}) = \begin{cases} x_1^N & \text{if } m = 1; \\ \left[x_1^N \;\; x_1^{N-1}\boldsymbol{p}_1(\tilde{\boldsymbol{x}})^T \;\; \cdots \;\; \boldsymbol{p}_N(\tilde{\boldsymbol{x}})^T\right]^T & \text{else} \end{cases} \tag{2}$$

[1] Published under LGPL-2.1: https://github.com/pwpfit.
[2] MATLAB's *smoothing spline* option for the built-in curve fitting function, for example, uses by default the observation points itself.
[3] https://mathworks.com/help/curvefit/fit.html

with $\tilde{\boldsymbol{x}} = (x_2, \dots, x_m)$ for $m > 1$, we can write

$$\boldsymbol{\mathcal{P}}_n(\boldsymbol{x}) = \begin{bmatrix} 1 & \boldsymbol{p}_1(\boldsymbol{x})^T & \cdots & \boldsymbol{p}_n(\boldsymbol{x})^T \end{bmatrix}^T. \tag{3}$$

By this notation, a polynomial f is expressed as scalar product of its monomials and coefficients,

$$f(\boldsymbol{x}) = \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q} \rangle \tag{4}$$

with the *vector of coefficients* $\boldsymbol{q}^T = \begin{bmatrix} b_1 & \cdots & b_{\mathfrak{r}[n]} \end{bmatrix}$.

2.2 Polynomial fitting

The observations $(\boldsymbol{x}_i, z_i)$ are conveniently given as sequences over $i \in [1, k]$:

Problem 2. Consider the k observations

$$z_i = \gamma(\boldsymbol{x}_i) + \epsilon_i, \tag{5}$$

where $(\boldsymbol{x}_i, z_i, \epsilon_i)_{1 \le i \le k} \subset \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}$ and $\gamma(\cdot)$ and $(\epsilon_i)_i$ are an unknown function and measurement error, respectively; find coefficients for $f = \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q} \rangle$ minimizing the *goodness of fit* (GoF)

$$\operatorname{GoF}(f) \stackrel{\text{def}}{=} \sum_{i=1}^{k} |f(\boldsymbol{x}_i) - z_i|^2 \tag{6}$$

for an $n > 0$.

Re-writing the goodness of fit using matrix calculus, we reduce the cost functional to a cost function and polynomial data fitting to a linear least-square problem.

Definition 3. A *linear least-square* (LSQ) problem is given as the optimization problem

$$\operatorname{lsq}(\mathbf{C}, \boldsymbol{d}) = \arg\min_{\boldsymbol{q}} \|\mathbf{C}\boldsymbol{q} - \boldsymbol{d}\|_2^2 \tag{7}$$

with $\boldsymbol{q} \in \mathbb{R}^r$, $\mathbf{C} \in \mathbb{R}^{k \times r}$, and $\boldsymbol{d} \in \mathbb{R}^k$.

We have the residuals in vector notation as

$$\boldsymbol{e} = \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{P}}_n(x_{1,1}, \dots, x_{1,m})^T \\ \vdots \\ \boldsymbol{\mathcal{P}}_n(x_{k,1}, \dots, x_{k,m})^T \end{bmatrix}}_{\stackrel{\text{def}}{=} \mathbf{K}} \boldsymbol{q} - \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix}}_{\stackrel{\text{def}}{=} \boldsymbol{\kappa}} \tag{8}$$

and the goodness of fit

$$\operatorname{GoF}(\boldsymbol{q}) = \|\boldsymbol{e}\|_2^2. \tag{9}$$

The coefficients of the optimal fit $\langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_0 \rangle$ now are subject to the linear-least square problem

$$\boldsymbol{q}_0 = \arg\min_{\boldsymbol{q}} \|\mathbf{K}\boldsymbol{q} - \boldsymbol{\kappa}\|_2^2. \tag{10}$$

3. PIECE-WISE FITTING

Problem 4. Take the observations of Problem 2; find coefficients $\boldsymbol{q}_1, \boldsymbol{q}_2$ such that

$$f \colon \boldsymbol{x} \longmapsto \begin{cases} \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_1 \rangle & \text{if } \varphi(\boldsymbol{x}) \le x_0; \\ \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_2 \rangle & \text{else} \end{cases}$$

with $\varphi \colon \mathbb{R}^m \to \mathbb{R}$ and $x_0 \in \mathbb{R}$ minimizes the goodness of fit of (6). [4]

We note the sub-polynomials of f by $f_{1,2} \colon \mathcal{X}_{1,2} \to \mathbb{R}, \boldsymbol{x} \mapsto \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_{1,2} \rangle$ and call $\mathcal{X}_1 \cup \mathcal{X}_2$ the *entire domain of* f. The *joint* of f is given as $\Omega_\varphi \stackrel{\text{def}}{=} \mathcal{X}_1 \cap \mathcal{X}_2 = \{\boldsymbol{x} \,|\, \varphi(\boldsymbol{x}) = x_0\}$.

[4] While solutions for multiple pieces can be derived, we focus on a single joint here.

The cost functional for f can be evaluated piece-wise to

$$\operatorname{GoF}(f) = \sum_{\boldsymbol{x}_i \in X_1} |f_1(\boldsymbol{x}_i) - z_i|^2 + \sum_{\boldsymbol{x}_i \in X_2} |f_2(\boldsymbol{x}_i) - z_i|_2^2, \tag{11}$$

where $X_1 = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{i'}\}$, $X_2 = \{\boldsymbol{x}_{i'+1}, \dots, \boldsymbol{x}_k\}$ are initial guesses of the subdomains.

We then have the residuals as $\boldsymbol{e}_{1,2} = \mathbf{K}_{1,2}\, \boldsymbol{q}_{1,2} - \boldsymbol{\kappa}_{1,2}$ with

$$\mathbf{K}_1 = \begin{bmatrix} \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}_{i'})^T \end{bmatrix}, \qquad \boldsymbol{\kappa}_1 = \begin{bmatrix} z_1 \\ \vdots \\ z_{i'} \end{bmatrix}; \tag{12}$$

$$\mathbf{K}_2 = \begin{bmatrix} \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}_{i'+1})^T \\ \vdots \\ \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}_k)^T \end{bmatrix}, \quad \boldsymbol{\kappa}_2 = \begin{bmatrix} z_{i'+1} \\ \vdots \\ z_k \end{bmatrix}; \tag{13}$$

and

$$\operatorname{GoF}(f) = \|\boldsymbol{e}_1\|_2^2 + \|\boldsymbol{e}_2\|_2^2 = \left\| \left[\, \boldsymbol{e}_1^T \,\vdots\, \boldsymbol{e}_2^T \,\right] \right\|_2^2. \tag{14}$$

Again, we reduce piece-wise fitting to the linear least-square problem

$$\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \arg\min_{\boldsymbol{q}'} \left\| \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_2 \end{bmatrix} \boldsymbol{q}' - \begin{bmatrix} \boldsymbol{\kappa}_1 \\ \boldsymbol{\kappa}_2 \end{bmatrix} \right\|_2^2 \tag{15}$$

with the *objective matrix* $\mathbf{K} \stackrel{\text{def}}{=} \operatorname{diag}(\mathbf{K}_1, \mathbf{K}_2)$.

Continuity of the piece-wise defined f over its entire domain holds if

$$\forall \boldsymbol{x} \in \Omega_\varphi. \quad \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_1 \rangle = \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_2 \rangle. \tag{16}$$

For single-variable functions, we have continuity for the identity function $\varphi = \mathrm{id}$ and x_0 is zero of

$$\langle \boldsymbol{\mathcal{P}}_n(x), \boldsymbol{q}_1 - \boldsymbol{q}_2 \rangle.$$

In the multivariate case, computing φ is generally hard.

4. CONSTRAINTS

To impose constraints on the coefficients (and thus the polynomials), we recall the *constrained* linear least-square problem (Haskell and Hanson, 1981)

$$\operatorname{lsq}(\mathbf{C}, \boldsymbol{d}, \mathbf{A}, \mathbf{0}) = \arg\min_{\boldsymbol{q} \in \Omega_{\mathbf{A}}} \|\mathbf{C}\boldsymbol{q} - \boldsymbol{d}\|_2^2. \tag{17}$$

with $\Omega_{\mathbf{A}} = \{\boldsymbol{q} \,|\, \mathbf{A}\boldsymbol{q} = \mathbf{0}\}$.

Lemma 5. Let $f_{1,2} = \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_{1,2} \rangle$ be polynomials; we have $f_1(\boldsymbol{x}) = f_2(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^{\mathfrak{r}[n]}$ *if and only if* $\boldsymbol{q}_1 = \boldsymbol{q}_2$.

In case of multiple variables or outputs, one may have x_0 for the single-variable, single-output case and ensure continuity in x_0 for all other variables and outputs.

Proposition 6. (Constraint of continuity). Let

$$\varphi(\boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{x} \le x_0 \tag{18}$$

be a linear matrix inequality (LMI) with $\boldsymbol{a}^T = [a_1 \;\cdots\; a_m]$ and $a_1 \neq 0$; a piece-wise polynomial function f with continuity in Ω_ϕ is subject to the constrained LSQ problem with continuity constraint matrix $\boldsymbol{\mathcal{C}}$.

Proof. We can simplify (18) to $\varphi(\boldsymbol{x}) = x_1 \le x_0$ w.l.o.g.:

Lemma 7. Let $\varphi \colon \boldsymbol{x} \mapsto \boldsymbol{a}^T \boldsymbol{x}$ with $a_1 \neq 0$; there is a linear, invertible $\boldsymbol{\pi}$ such that

$$(\varphi \circ \boldsymbol{\pi}) \colon \boldsymbol{y} \longmapsto y_1 \tag{19}$$

with $\boldsymbol{y} = (y_1, \dots, y_m)$.

For $\varphi(\boldsymbol{x}) \neq x_1$, we thus fit polynomials $g_{1,2}$ to $(\boldsymbol{\pi}\boldsymbol{x}_i, z_i)_i$ such that $g_{1,2}$ join in $(\varphi \circ \boldsymbol{\pi})(\boldsymbol{y}) = x_0$ and find $f_{1,2}$ as

$$f_1 = \left(g_1 \circ \boldsymbol{\pi}^{-1}\right); \qquad f_2 = \left(g_2 \circ \boldsymbol{\pi}^{-1}\right). \tag{20}$$

We now have continuity if

$$\forall \boldsymbol{x} \in \Omega_{x_0}. \quad \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_1 \rangle = \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_2 \rangle \tag{21}$$

with $\Omega_{x_0} = \{\boldsymbol{x} \,|\, x_1 = x_0\}$; hence

$$\forall \tilde{\boldsymbol{x}} \in \mathbb{R}^{m-1}. \quad \langle \boldsymbol{\mathcal{P}}_n(x_0, \tilde{\boldsymbol{x}}), \boldsymbol{q}_1 \rangle = \langle \boldsymbol{\mathcal{P}}_n(x_0, \tilde{\boldsymbol{x}}), \boldsymbol{q}_2 \rangle. \tag{22}$$

Separation of the assigned variable $x_1 \equiv x_0$ as $\boldsymbol{\Lambda}_0^T$ yields

$$\begin{aligned} \langle \boldsymbol{\mathcal{P}}_n(x_0, \tilde{\boldsymbol{x}}), \boldsymbol{q}_{1,2} \rangle &= \langle \boldsymbol{\Lambda}_0^T \boldsymbol{\mathcal{P}}_n(\tilde{\boldsymbol{x}}), \boldsymbol{q}_{1,2} \rangle \\ &= \langle \boldsymbol{\mathcal{P}}_n(\tilde{\boldsymbol{x}}), \boldsymbol{\Lambda}_0 \boldsymbol{q}_{1,2} \rangle \end{aligned} \tag{23}$$

with

$$\boldsymbol{\Lambda}_0 = \left[\begin{array}{c|c|c|cccc} 1 & x_0 & & x_0^n & & & \\ & \operatorname{diag} \boldsymbol{p}_1(\boldsymbol{1}_{m-1}) & \cdots & & x_0^{n-1} \operatorname{diag} \boldsymbol{p}_1(\boldsymbol{1}_{m-1}) & & \\ \hline & & & & & \ddots & \\ & & & & & & \operatorname{diag} \boldsymbol{p}_n(\boldsymbol{1}_{m-1}) \end{array}\right], \tag{24}$$

where $\boldsymbol{1}_{m-1} \in \{1\}^{m-1}$. By Lemma 5, we have that

$$\langle \boldsymbol{\mathcal{P}}_n(\tilde{\boldsymbol{x}}), \boldsymbol{\Lambda}_0 \boldsymbol{q}_1 \rangle = \langle \boldsymbol{\mathcal{P}}_n(\tilde{\boldsymbol{x}}), \boldsymbol{\Lambda}_0 \boldsymbol{q}_2 \rangle \tag{25}$$

for all $\tilde{\boldsymbol{x}} \in \mathbb{R}^{m-1}$ if and only if $\boldsymbol{\Lambda}_0 \boldsymbol{q}_1 = \boldsymbol{\Lambda}_0 \boldsymbol{q}_2$. Hence, the constraint of continuity is written as

$$\left[\, \boldsymbol{\Lambda}_0 \;\vdots\; -\boldsymbol{\Lambda}_0 \,\right] \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \boldsymbol{0} \tag{26}$$

and $\boldsymbol{\mathcal{C}} = [\boldsymbol{\Lambda}_0 \;\; -\boldsymbol{\Lambda}_0]$.

Due to measurement errors or modelling flaws, a polynomial fitting may have relations that shall not be modeled;[5] in this case, it is desirable to constrain the resulting polynomial to be zero (or constant) for certain parameters $\tilde{\boldsymbol{x}}^* = (x_{j+1}, \cdots, x_m)$:

Proposition 8. (Zero constraint). Let $\boldsymbol{x}^* = (x_1, \ldots, x_j)$ for $j > 0$; a polynomial $f = \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q} \rangle$ with

$$\forall \boldsymbol{x}^* \in \mathbb{R}^j. \quad \langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}^*, \boldsymbol{0}_{m-j}), \boldsymbol{q} \rangle = 0 \tag{27}$$

with $\boldsymbol{0}_{m-j} \in \{0\}^{m-j}$ is subject to the zero constraint matrix $\boldsymbol{\mathcal{Z}}$.

Proof. Separating the assigned parameters $\tilde{\boldsymbol{x}}^* = \boldsymbol{0}_{m-j}$ as $\mathbf{V}_0^T$ and applying Lemma 5, we have that

$$\langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}^*), \mathbf{V}_0 \boldsymbol{q} \rangle = 0 \tag{28}$$

for all $\boldsymbol{x}^* \in \mathbb{R}^j$ if and only if $\mathbf{V}_0 \boldsymbol{q} = \boldsymbol{0}$.

Using $\mathbf{V}' = \operatorname{diag}\left(v_1, \ldots, v_{\mathfrak{r}[n]}\right)$ where $v_i = 1$ if the i-th element of $\boldsymbol{\mathcal{P}}_n(\boldsymbol{1}_j, \boldsymbol{0}_{m-j})$ is non-zero, $v_i = 0$ otherwise, $\mathbf{V}_0$ is obtained by removing the all-zero rows of $\mathbf{V}'$, thus ensuring full rank.

For piece-wise polynomial fitting with zero constraint, take

$$\boldsymbol{\mathcal{Z}} = \left[\begin{array}{c|c} \mathbf{V}_0 & \boldsymbol{0} \\ \hline \boldsymbol{0} & \mathbf{V}_0 \end{array}\right]. \tag{29}$$

If both zero constraint and constraint of continuity are given, we need to ensure full rank of the complete constraint matrix:

$$\begin{bmatrix} \boldsymbol{\mathcal{C}} \\ \boldsymbol{\mathcal{Z}} \end{bmatrix} \boldsymbol{q}' = \left[\begin{array}{c|c} \boldsymbol{\Lambda}_0 & -\boldsymbol{\Lambda}_0 \\ \mathbf{V}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \mathbf{V}_0 \end{array}\right] \begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \boldsymbol{0}.$$

[5] E.g., for a symmetric aircraft aligned to the flow, there is no side-force—regardless its angle of attack.

5. IMPLEMENTATION

The `pwpfit` toolbox is implemented in MATLAB using the *Optimization toolbox*[6] for linear least-square solving and *Symbolic math toolbox*[7] for representation of the vector of monomials.

As MATLAB is rather slow on arrays of variable length, we use a statically allocated array to generate the vector of monomials $\boldsymbol{\mathcal{P}}_n$ in m variables. Applying a recursive sub-routine (Alg. 1) to write the auxiliary $\boldsymbol{p}_N(\boldsymbol{x})$ at the `l`-th(and following) positions of `P`, the vector of monomials is then computed as *symbolic expression* `P` of parameters `X:=`$\boldsymbol{x}$ according to (3).

The length of `P`, *i.e.* the number of monomials in $\boldsymbol{\mathcal{P}}_n(\boldsymbol{x})$, is given as sum of multicombinations

$$\mathfrak{r}[n] = \sum_{N=1}^{n} \binom{m+N-1}{N-1} = \binom{m+n}{n}. \tag{30}$$

Algorithm 1. Recursive algorithm for $\boldsymbol{p}_N(\boldsymbol{x})$.

```
function [P,l] = MONOMIAL(P,l,X,m,n,X0=1)
    if m == 1 then
        P(l) = X0*X^n;
        l = l+1;
    else
        for j = 0:n
            X0 = X0*X(1)^(n-j));
            [P,l] = ...
            MONOMIAL(P,l,X(2:end),m-1,j,X0);
        end
    end
end
```

Alg. 2 illustrates the computation of the left-hand side of the continuity constraint matrix, $\boldsymbol{\Lambda}_0$, for $\varphi(\boldsymbol{x}) = x_1 \leq x_0$, using the auxiliary vectors $\boldsymbol{p}_N(x_0, \boldsymbol{1}_{m-1})$ in degrees $N \in [0, n]$ with $\boldsymbol{1}_{m-1} \sim$ `one`.

Algorithm 2. Code-snippet for $\boldsymbol{\Lambda}_0 \sim$ `Aeq` in x_0.

```
one = num2cell(ones(1,m-1));
j = 0;
for N=0:n
  % let pN:=p_N(·); rN:=r[N]
    pNx0 = double(pN(x0,one{:}));
    Aeq(1:rN,j+(1:rN)) = diag(pNx0);
    j = j + rN;
end
```

Given a vector `y0` whose i-th component is zero if and only if the fitted polynomials are zero in the parameter x_i, Alg. 3 yields the zero separation matrix $\mathbf{V}_0$. Here, we make direct use of MATLAB's logical indexing for matrices in order to remove the all-zero rows of the square matrix $\mathbf{V}'$.

[6] https://mathworks.com/help/optim

[7] https://mathworks.com/help/symbolic

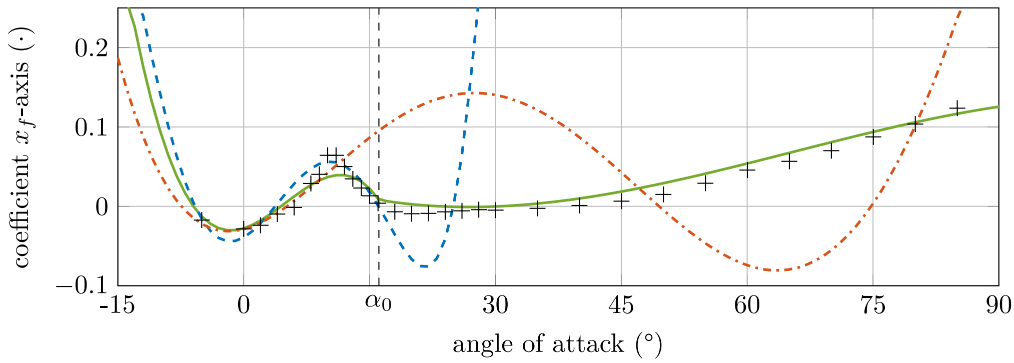


Fig. 1. Observed coefficients $\hat{C}_X(\hat{\alpha})$ (+) and comparison of 3rd-order polynomial (- - -, -·-·-) and piece-wise (——) identifications. (Cunis et al., 2018)

Algorithm 3. Code-snippet for $\mathbf{V}_0 \sim$ `Azero`.

```
% let p:=𝓟_n(·); r:=𝔯[n]
Azero = eye(r);
Y = num2cell(y0);
pY = double(p(Y{:}));
Azero(pY==0,:) = [];
```

The constrained linear least-square problem is solved by the `lsqlin` function of the Optimization toolbox. As `lsqlin` requires a linear inequality constraint,

$$\mathbf{A}\boldsymbol{q} \leq b,$$

we assign $\mathbf{A} = [1 \ \cdots \ 1]$ and $b = 10^4$.

If no continuity constraints are given, the joint x_0 of a single-variable function with $\varphi = \mathrm{id}$ is found using a non-linear function solver.[8] The resulting coefficients and their joint are returned as `pwfitobject`, which provides interfaces for plotting and exporting the obtained piece-wise function and the polynomial sub-functions.

The auxiliary functions `prepareHyperSurfaceData` and `LMI2single` are provided to prepare tabular data for fitting[9] and to simplify an LMI constraint of continuity (Lemma 7), respectively.

6. AERODYNAMIC IDENTIFICATION

The aerodynamic coefficients of an aircraft are subject to, amongst others, its angle of attack, side-slip angle, the deflection of ailerons, elevator, and rudder, as well as the body rates. Measurements for various inputs, e.g. of the NASA Generic Transport Model (GTM, Jordan et al., 2006), are usually performed in the wind-tunnel:

Example 9. (GTM[10]). The observations of the aerodynamic coefficients of the GTM are given by the unknown function $\Gamma(\cdot)$ to

$$\hat{\boldsymbol{C}} = \Gamma\left(\hat{\alpha}, \hat{\beta}, \hat{\xi}, \hat{\eta}, \hat{\zeta}\right) + \boldsymbol{\epsilon} \tag{31}$$

for the observed inputs $\hat{\alpha} \in A$, $\hat{\beta} \in B$, $\hat{\xi} \in \Xi$, $\hat{\eta} \in H$, and $\hat{\zeta} \in Z$ with $\hat{\boldsymbol{C}} = \left(\hat{C}_{\mathrm{X}}, \hat{C}_{\mathrm{Y}}, \hat{C}_{\mathrm{Z}}, \hat{C}_{\mathrm{l}}, \hat{C}_{\mathrm{m}}, \hat{C}_{\mathrm{n}}\right)$ and $\boldsymbol{\epsilon}$ an unknown measurement error.

For polynomial and piece-wise polynomial fitting, observations in (31) have to be transformed to tabular data

$$(\boldsymbol{C}_i)_{1\leq i\leq k} = \Gamma(A \times B \times \Xi \times H \times Z) + (\boldsymbol{\epsilon}_i)_{1\leq i\leq k} \tag{32}$$

with $\boldsymbol{C}_i = (C_{\mathrm{X},i}, C_{\mathrm{Y},i}, C_{\mathrm{Z},i}, C_{\mathrm{l},i}, C_{\mathrm{m},i}, C_{\mathrm{n},i})$ and

$$k = |A \times B \times \Xi \times H \times Z|\,. \tag{33}$$

Here, simple polynomials models seem unsuitable to represent the full-envelope aerodynamics (Fig. 1; see also Cunis et al., 2018). At the stall angle of attack, the laminar flow around the wings of the pre-stall region changes to turbulent flow and remains so in post-stall. This significant change of the flow dynamics motivates a piece-wise fitting of the pre-stall and post-stall dynamics:[11]

$$C_{\odot}(\alpha,\beta,\xi,\eta,\zeta) = \begin{cases} C_{\odot}^{pre}(\alpha,\beta,\xi,\eta,\zeta) & \text{if } \alpha \leq \alpha_0 \\ C_{\odot}^{post}(\alpha,\beta,\xi,\eta,\zeta) & \text{else} \end{cases}$$

where $C_{\odot} \in \{C_{\mathrm{X}}, C_{\mathrm{Y}}, C_{\mathrm{Z}}, C_{\mathrm{l}}, C_{\mathrm{m}}, C_{\mathrm{n}}\}$ are 6-dimensional polynomials. Initially, $\alpha_0 = 16.11°$ is found by fitting $C_{\mathrm{X}\alpha}$ with respect to the angle of attack only, resulting in

$$C_{\mathrm{X}\alpha}^{pre}(\alpha_0) = C_{\mathrm{X}\alpha}^{post}(\alpha_0)\,,$$

which is the boundary angle of attack. The boundary condition $\alpha \equiv \alpha_0$ then resembles a 5-dimensional hyperplane.

We now have continuity of the coefficient functions over their entire domain if

$$C_{\odot}^{pre}(\alpha_0,\cdots) \equiv C_{\odot}^{post}(\alpha_0,\cdots)\,.$$

At last, we require the lateral coefficients (C_{Y}, C_{l}, C_{n}) to vanish in the symmetric setting, *i.e.* zero side-slip, no aileron nor rudder deflection ($\beta = \xi = \zeta = 0$).

The obtained, piece-wise polynomial models for the C_{X} and C_{Y} coefficients are exemplary shown in Fig. 2 for angle of attack and side-slip angle with neutral surface deflections ($\xi = \eta = \zeta = 0$). Besides, the residuals

$$e_{\mathrm{X}} = C_{\mathrm{X}}\left(\hat{\alpha}, \hat{\beta}\right) - \hat{C}_{\mathrm{X}}$$

$$e_{\mathrm{Y}} = C_{\mathrm{Y}}\left(\hat{\alpha}, \hat{\beta}\right) - \hat{C}_{\mathrm{Y}}$$

are given for $\left(\hat{\alpha}, \hat{\beta}\right) \in A \times B$.

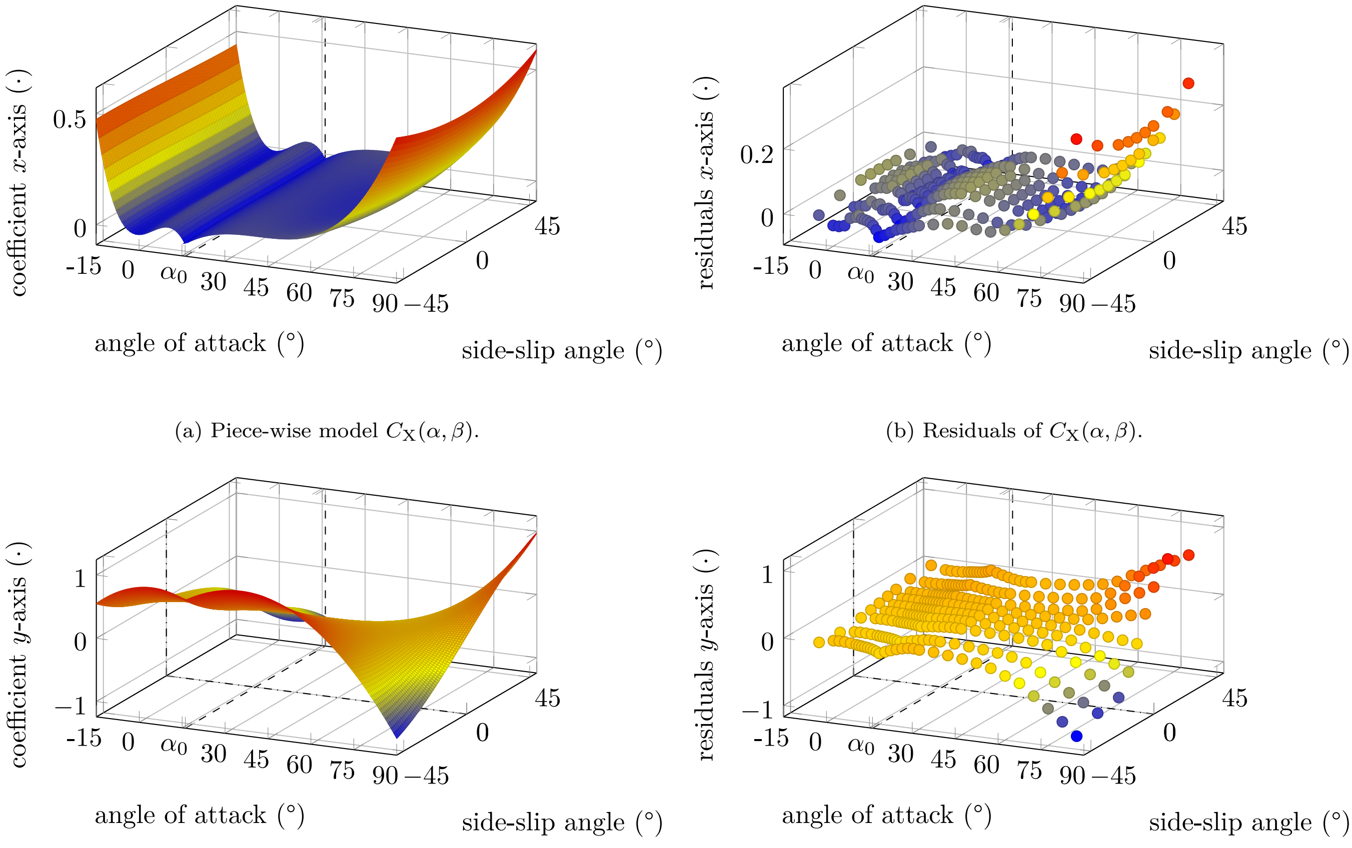
A six-degrees-of-freedom trim analysis of the GTM with piece-wise polynomial, aerodynamic coefficients has been presented in (Cunis et al., 2017).

[8] https://mathworks.com/help/optim/ug/fsolve.html

[9] Extending MATLAB's functions `prepareCurveData` and `prepareSurfaceData`.

[10] https://software.nasa.gov/software/LAR-17625-1

[11] A script for MATLAB can be found in the `demo` folder.



(a) Piece-wise model $C_{\mathrm{X}}(\alpha, \beta)$.

(b) Residuals of $C_{\mathrm{X}}(\alpha, \beta)$.

(c) Piece-wise model $C_{\mathrm{Y}}(\alpha, \beta)$.

(d) Residuals of $C_{\mathrm{Y}}(\alpha, \beta)$.

Fig. 2. Piece-wise model of the C_{X} and C_{Y} coefficients of the Generic Transport Model in angle of attack α and side-slip angle β, and their residuals; for surface deflections $\xi = \eta = \zeta = 0$. Both models are continuous in the joint $\alpha \equiv \alpha_0$ and the lateral C_{Y} model vanishes in $\beta \equiv 0$.

7. NOTE ON COMPUTATION TIME

When fitting polynomials of high dimension to large data sets, the computation of a single polynomial in all variables usually takes a considerably long time. In Tab. 1 we compare the computation time for objective matrix, constraint matrices, and the solution of the resulting LSQ problem for all six coefficients of Example 9.
Here, the objective matrix $\mathbf{K}$ takes by far the most time; by (15), the size of $\mathbf{K}$ resolves to

$$2k \times \mathfrak{r}[n] \tag{34}$$

and both k (33) and $\mathfrak{r}[n]$ (30) grow exponentially with the number of variables m. The size of $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{Z}}$, too, grow with m but are independent of k. [12]

Rather than single, high-dimensional polynomials, it may be more appropriate to sequentially fit sums of polynomial terms lower dimensions, for sub-sets of the variables:

$$\begin{aligned} C_\odot = C_{\odot\alpha}(\alpha) + C_{\odot\beta}(\alpha, \beta) + C_{\odot\xi}(\alpha, \beta, \xi) \\ + C_{\odot\eta}(\alpha, \beta, \eta) + C_{\odot\zeta}(\alpha, \beta, \zeta) \end{aligned} \tag{35}$$

with $m \leq 3$. In this case, continuity of each term in α_0 implies continuity of $C_\odot$ over its entire domain. Tab. 2 shows the reduced computation time for the sequential fit of C_{X}.

[12]In addition, the computation of $\boldsymbol{\mathcal{Z}}$ by MATLAB's logical indexing is obviously very efficient.

Table 1. Time consumption for fit of multi-variate polynomials $C_\odot(\alpha, \beta, \xi, \eta, \zeta)$: computation time for objective matrix $\mathbf{K}$, continuity constraint matrix $\boldsymbol{\mathcal{C}}$, zero constraint matrix $\boldsymbol{\mathcal{Z}}$, and solving the LSQ problem. All values in seconds with accuracy $\pm 10\,\mathrm{ms}$ (Intel Core i7, 3 GHz, 16 GB).

		$\mathbf{K}$	$\boldsymbol{\mathcal{C}}$	$\boldsymbol{\mathcal{Z}}$	lsq
C_{X}	$(\alpha, \beta, \xi, \eta, \zeta)$	2058.39	0.84	—	2.60
C_{Y}	$(\alpha, \beta, \xi, \eta, \zeta)$	2100.64	0.59	—	2.62
C_{m}	$(\alpha, \beta, \xi, \eta, \zeta)$	2100.22	0.59	—	2.62
C_{Y}	$(\alpha, \beta, \xi, \eta, \zeta)$	2102.25	0.60	<0.01	1.71
C_{l}	$(\alpha, \beta, \xi, \eta, \zeta)$	2109.07	0.63	<0.01	1.65
C_{n}	$(\alpha, \beta, \xi, \eta, \zeta)$	2102.38	0.59	0.01	1.54

Table 2. Time consumption for sequential fit of polynomial sum $C_{\mathrm{X}} = C_{\mathrm{X}\alpha} + C_{\mathrm{X}\beta} + C_{\mathrm{X}\xi} + C_{\mathrm{X}\eta} + C_{\mathrm{X}\zeta}$: computation time for objective matrix $\mathbf{K}$, continuity constraint matrix $\boldsymbol{\mathcal{C}}$, and solving the LSQ problem. All values in seconds with accuracy $\pm 10\,\mathrm{ms}$ (Intel Core i7, 3 GHz, 16 GB).

		$\mathbf{K}$	$\boldsymbol{\mathcal{C}}$	lsq
$C_{\mathrm{X}\alpha}$	(α)	0.16	—	0.04
$C_{\mathrm{X}\beta}$	(α, β)	4.89	0.16	0.15
$C_{\mathrm{X}\xi}$	(α, β, ξ)	39.06	0.25	0.10
$C_{\mathrm{X}\eta}$	(α, β, η)	31.34	0.20	0.11
$C_{\mathrm{X}\zeta}$	(α, β, ζ)	36.49	0.20	0.04
C_{X}	$(\alpha, \beta, \xi, \eta, \zeta)$	111.94	0.81	0.44

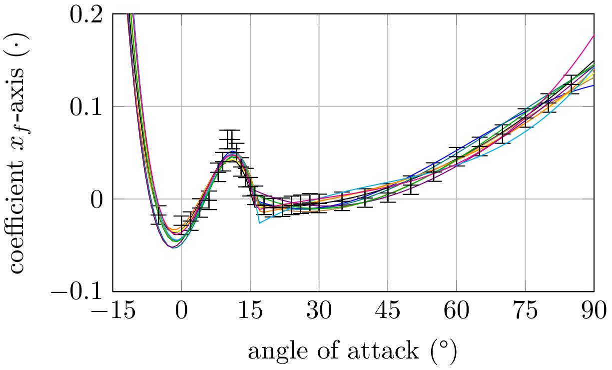


Fig. 3. Piece-wise fits of erroneous coefficients ($\sigma_X = 0.01$).

8. SENSITIVITY ANALYSIS

In order to study the sensitivity of piece-wise fitting, we take the GTM coefficients data of Example 9 as "true" values ($\epsilon \equiv 0$) and add a white noise ν_X:

$$C_{\mathrm{X}}^{\dagger} = \Gamma_X(\hat{\alpha}) + \nu_X \tag{36}$$

and ν_X is normally distributed with deviation $\sigma_X \stackrel{\text{def}}{=} 0.01$. We then compute a batch of piece-wise fits

$$\left(C_{\mathrm{X}}^{\{j\}}(\alpha)\right)_j$$

for 10 000 noise samples; a family of obtained curves is shown in Fig. 3.

The joints $\alpha_0^{\{j\}}$ have a sample mean $\overline{\alpha}_0 = 16.11°$ and deviation $\sigma_\alpha = 0.51°$. The error of fit with respect to the "true" values has a sample standard deviation

$$\sigma\left(C^{\{j\}}(\hat{\alpha}) - \Gamma_X(\hat{\alpha})\right) < \sigma_X$$

for all observations $\hat{\alpha}$. That is, piece-wise polynomial fitting is able to reduce the error with respect to the erroneous signal.

9. CONCLUSION

With the rise of multivariate splines, prior research to piece-wise polynomial regression has been abandoned. However, by pre-selection of the knots, spline fitting does not take into the underlying model; in fact, it thus overestimates the observations. On the other hand, the estimation of the "true" switching points of a piece-wise physical system usually adds computational difficulty and load.

In this paper, we have presented an approach of piece-wise polynomial fitting using the LSQ optimization technique in order to fit both polynomial models and the joint point. The `pwpfit` toolbox for MATLAB provides functions for polynomial and piece-wise polynomial data fitting under continuity and zero constraints. We demonstrated our approach by fitting piece-wise polynomial models of the aerodynamic coefficients of an airliner model; here, we argued that simple polynomial models are unsuitable for the full-envelope dynamics while the dynamical changes at the stall point prompt the application of piece-wise regression. By simulation of the sensitivity to random noise samples, we proved that piece-wise polynomial fitting improves the estimation of an erroneous signal.

REFERENCES

Chaudhuri, P., Huang, M.C., Loh, W.Y., and Yao, R. (1994). Piecewise-Polynomial Regression Trees. *Statistica Sinica*, 4(1), 143–167.

Cunis, T., Burlion, L., and Condomines, J.P. (2018). Piece-wise Identification and Analysis of the Aerodynamic Coefficients, Trim Conditions, and Safe Sets of the Generic Transport Model. In *AIAA Guidance, Navigation, and Control Conference*. Kissimmee, US-FL.

Cunis, T., Condomines, J.P., and Burlion, L. (2017). Full-envelope, Six-Degrees-of-Freedom Trim Analysis of Unmanned Aerial Systems based on Piece-wise Polynomial Aerodynamic Coefficients. In *4th Workshop on Research, Education, and Development of Unmanned Aerial Systems*. Linköping, SE.

de Visser, C.C., Chu, Q.P., and Mulder, J.A. (2009). A new approach to linear regression with multivariate splines. *Automatica*, 45(12), 2903–2909.

Gallant, A.R. and Fuller, W.A. (1973). Fitting segmented polynomial regression models whose join points have to be estimated. *Journal of the American Statistical Association*, 68(341), 144–147.

Golub, G.H. (1982). Numerical Methods for Solving Least Squares Problems. Technical report, U.S. Army Research Office.

Haskell, K.H. and Hanson, R.J. (1981). An Algorithm for Linear Least Squares Problems with Equality and Nonnegativity Constraints. *Mathematical Programming*, 21, 98–118.

Jordan, T.L., Foster, J.V., Bailey, R.M., and Belcastro, C.M. (2006). AirSTAR: A UAV Platform for Flight Dynamics and Control System Testing. In *AIAA Aerodynamics Measurement Technology and Ground Testing Conference*. San Francisco, US-CA.

Kariya, T. and Kurata, H. (2004). *Generalized Least Squares*. Wiley Series in Probability and Statistics. John Wiley & Sons, Chichester, GB.

Klein, V. and Morelli, E.A. (2006). *Aircraft System Identification: Theory and Practice*. American Institute of Aeronautics and Astronautics, Reston, US-VA.

Lawson, C.L. and Hanson, R.J. (1995). *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics.

McGee, V.E. and Carleton, W.T. (1970). Piecewise Regression. *Journal of the American Statistical Association*, 65(331), 1109–1124.

Robison, D.E. (1964). Estimates for the Points of Intersection of Two Polynomial Regressions. *Journal of the American Statistical Association*, 59(305), 214–224.

Appendix A. PROOFS

Proof. [Lemma 5] *By reduction to:*

$$\langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_1 - \boldsymbol{q}_2 \rangle \equiv 0 \Longleftrightarrow \boldsymbol{q_1} - \boldsymbol{q_2} = 0$$

where $\langle \boldsymbol{\mathcal{P}}_n(\boldsymbol{x}), \boldsymbol{q}_1 - \boldsymbol{q}_2 \rangle$ is the zero polynomial.

Proof. [Lemma 7] *By construction:*

$$\boldsymbol{\pi}^{-1} = \begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix};$$

$\boldsymbol{\pi}^{-1}$ is invertible as $\left|\boldsymbol{\pi}^{-1}\right| = a_1$ and $\varphi(\boldsymbol{x}) = y_1 \Leftrightarrow \boldsymbol{x} = \boldsymbol{\pi}\boldsymbol{y}$.